



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/693,749

10/24/2003

Thekkthalackal Varugis Kurien

MSFT-2839/301118.1

2449

41505

7590

04/30/2008

WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)  
CIRA CENTRE, 12TH FLOOR  
2929 ARCH STREET  
PHILADELPHIA, PA 19104-2891

EXAMINER

LE, CANH

ART UNIT

PAPER NUMBER

2139

MAIL DATE

DELIVERY MODE

04/30/2008

PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/693,749	<b>Applicant(s)</b> KURIEN ET AL.	
	<b>Examiner</b> CANH LE	<b>Art Unit</b> 2139	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 25 February 2008.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-34 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-34 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)          | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____                                      |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____  | 6) <input type="checkbox"/> Other: _____                          |

### **DETAILED ACTION**

This Office Action is in response to the application filed on 02/25/2008.

Claims 1, 13, 25, and 32 have been amended.

Claims 1-34 have been examined and are pending.

#### ***Continued Examination Under 37 CFR 1.114***

A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 02/25/2008 has been entered.

#### ***Response to Amendment***

The applicant's amendment filed 11/26/2007 necessitated the new ground(s) of rejection presented in this Office action. Therefore, applicant's arguments with respect to claims have been considered but are moot in view of the new ground(s) of rejection.

#### ***Specification***

The disclosure is objected to because of the following informalities: In the specification, paragraph [0056], it recites "**base component 506**". There is no base component 506 in the figure 6. Appropriate correction is required.

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: There is no antecedent basis for “**a base layer**” for claim 1.

***Claim Rejections - 35 USC § 101***

35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

**Claims 1-12 and 32-34** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

**Claim 1** is directed to a system that manages the partition of an application comprising a base layer. The base layer appears to be a software layer. Software is non-statutory.

**Claims 32-34** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Claim 32 is directed to “a system that supports the partitioning of an application into at least a first software object and a second software object, the system hosting a first environment and a second environment... comprising an **application program interface**”. The application program interface is software. Software is non-statutory.

Claims 2-12 are rejected due to virtual dependency of claim 1.

Claims 33-34 are rejected due to virtual dependency of claim 32.

***Claim Rejections - 35 USC § 112***

The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

**Claims 1-13, 16-18, 20, and 32-34** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

**Claims 1, 8, and 12** recites the limitation “**base layer**” in lines 1-2. It is vague. The term is indefinite because the specification does not clearly define. The Examiner interprets “base layer” as “base component 508” in figures 5-6 and paragraph [0056] of the specification.

Claims 2-12 are rejected due to virtual dependency of claim 1.

**Claim 6** recites the limitation “second application” in line 1. There is insufficient antecedent basis for this limitation in the claim.

**Claim 6** recites the limitation “second application” in line 1. It is vague. Does the “second application” means a “second software object” in claim 5 or claim 1? There is no “second application” in claims 1 and 5.

**Claim 13** recites the limitation “an assurance policy” in line 1. It is vague. The term is indefinite because the specification does not clearly define. There is no support for assurance policy or policy in the specification. The Examiner interprets “an

assurance policy” as “assurance environment” in figures 5-6 and paragraph in the specification.

The Examiner respectfully requests the Applicant to identify where "an assurance policy" is in the specification.

**Claim 18 and 16** recite the limitation “first software object **causes** a representation ” in line 1. It is vague how the first software object causes a representation of plurality of data.

**Claim 18** recites the limitation “to **permit** viewing of image of the data” in line 4. This does not prohibit a computer from doing the recited acts. They do not cause any functionality to occur in the computer. It's unclear what Applicant's intended metes and bounds of the claim are, since the claim appears to cover anything and everything that does not prohibit actions from occurring.

Claims 17-18 are rejected due to virtual dependency of claim 16.

**Claim 20** recites the limitation "policy" in line 1. There is insufficient antecedent basis for this limitation in the claim.

**Claim 20** recites the limitation "policy" in line 1. It is vague. The term is indefinite because the specification does not clearly define. There is no support for assurance policy or policy in the specification.

**Claim 32** recites “a **system** that supports the partitioning of an application into at least a first software object and a second software object, the system hosting a first

environment and a second environment, the first software object running in the first environment, the second software object running in the second environment, the system comprising an application programming interface that exposes **at least one of the following methods**". It is unclear claim 32 is the system claim or the method claim.

Claim 32 recites "the partitioning" in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claims 33-34 are rejected due to virtual dependency of claim 32.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

**Claims 1, 5-14, 19-20, 23-26, and 31-34** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **J.E. Smith**, "An Overview of Virtual Machine Architectures", October 27, 2001, pp. 1-20.

#### **As per claim 1:**

Muschellack teaches a system that manages the partitioning of an application comprising: a base layer [**fig. 8, hardware components**] that hosts the operation of a

first environment **[fig. 8, Standard Mode 430]** and a second environment **[fig. 8, Nexus Mode 432]**, the application comprising:

(a) a first software object of said application that executes in said first environment comprising a first operating system, said first software object handling a plurality of data and including logic to identify a first of said plurality of data as not processable by said first software object **[fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430];**

(b) and a second software object of said application that executes in said second environment comprising a second operating system and that processes said first of said plurality of data in a manner that resists tampering with said first of said plurality of data **[fig. 8; Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432)];**

(c) said base layer comprising or hosting logic that receives said first of said plurality of data from said software object and routes said first of said plurality of data to said second environment, such that functionality of the application is parsed between the first and second operating systems **[Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]**.



Muschellack does not explicitly disclose the base layer acting as a virtual machine.

However, in an analogous art, Smith teaches an overview of Virtual Machine Architectures, wherein a base layer that hosts the operation of a first environment and a second environment the application **[pg. 1-9; fig. 3-4]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Smith, wherein a base layer that hosts the operation of a first environment and a second environment the application to allow interoperability of the major system components such as different operating system. Multiple Operating System environment can co-exist on the same computer, in strong isolation from each other. A self-contained operating environment that behaves as if it is a separate computer **[Smith, pg. 6, 2<sup>nd</sup> paragraph]**. The virtual machine offers greater portability as well as robustness and reliability.

**As per claim 5:**

Muschellack and Smith teach the system of claim1. Muschellack further teaches said first of said plurality of said is entered on a keyboard **[fig. keyboard 16]**, and wherein the resistant to tampering provided by said second software object comprises resisting tampering with said first of said plurality of data in transit from said keyboard to an input stream of said second software object **[fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430**

**(fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"].**

**As per claim 6:**

Muschellack further teaches the system of claim 5, wherein said second application signs said first of said plurality of data to prevent subsequent tampering with said first of said plurality of data **[Col. 9, line 60 to Col. 10, line 3; "The components (i.e. application) may include or have access to applications which provide cryptographic functions for performing, encryption, decryption, digital signature signing, digital signature verification, hashing and/or other cryptographic calculation...a secure communication session between components"]**.

**As per claim 7:**

Muschellack further teaches the system of claim 6, wherein said second environment signs said first of said plurality of data and the signature created by said second application as an indication that said first of said plurality of data and said signature were created in said second environment **[Col. 9, line 60 to Col. 10, line 3; "The components (i.e. application) may include or have access to applications which provide cryptographic functions for performing, encryption, decryption, digital signature signing, digital signature verification, hashing and/or other**

**cryptographic calculation...a secure communication session between components”].**

**As per claim 8:**

Muschellack and Smith teach the system of claim 1. Smith further teaches an system, wherein a base layer comprises a component that assigns a first identifier to said second environment **[pg. 1-9; fig. 3-4; Virtual machine can be used to connect the major system components; translating from one instruction set to another; optimizing an existing application binary for the same instruction set; replicating a virtual machine so that multiple (possible different) OSes can be supported simultaneously; composing virtual machine software to form a more complex, flexible system”; Virtual machine controls different Operating System in different environments; The virtual machine supports multiple operating systems by assigning different identifiers and software objects in different environments].**

**As per claim 9:**

Muschellack and Smith teach the system of claim 8. Smith’s virtual machine further encompasses said first of said plurality of data includes, or is accompanied by, said first identifier and a second identifier that identifies said second software object **[pg. 1-9; fig. 3-4; Virtual machine can be used to connect the major system components; translating from one instruction set to another; optimizing an existing application binary for the same instruction set; replicating a virtual**

**machine so that multiple (possible different) OSes can be supported simultaneously; composing virtual machine software to form a more complex, flexible system"; Virtual machine controls different Operating System in different environments; The virtual machine supports multiple operating systems by assigning different identifiers and software objects in different environments].**

**As per claim 10:**

Muschellack and Smith teach the system of claim 1. Muschellack further teaches said first environment is associated with a first specification that describes the behavior of said first environment, wherein said second environment is associated with a second specification that describes the behavior of said second environment, wherein there is a higher level of assurance that said second environment will conform to said second specification than that said first environment will conform to said first specification **[fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]**.

**As per claim 11:**

Muschellack further teaches the system of claim 10, wherein said second software object relies upon the behavior of the second environment in order to resist tampering with said first of said plurality of data **[fig. 8, Nexus Mode 432]**.

**As per claim 12:**

Muschellack and Smith teach the system of claim 1. Smith further teaches said base layer is said second environment, or is included within said second environment **[fig. 3-4, virtual machine]**.

**As per claim 13:**

Muschellack teaches a method of a first software object of an application, which executes in a first environment comprising a first operating system **[fig. 8; standard mode Operating system]**, handling data to which an assurance policy applies, the method comprising:

(a) the first software object encountering the data **[fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430]**;

(b) the first software object determining that the data is not processable by the first software object **[fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430]**;

(c) the first software object causing the data to be provided to a second software object of the application that executes in a second environment comprising a second operating system, the second environment providing a first level of assurance that actions performed in the second environment will be performed correctly, wherein the

second software object processes the data in a manner that uses said assurance policy to create resistance to tampering with the data by acts arising outside of the second environment, such that functionality of the application is parsed between the first and second operating systems **[Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]**.

Muschellack does not explicitly disclose the base layer acting as a virtual machine.

However, in an analogous art, Smith teaches an overview of Virtual Machine Architectures, wherein a base layer that hosts the operation of a first environment and a second environment the application **[pg. 1-9; fig. 3-4]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Smith, wherein a base layer that hosts the operation of a first environment and a second environment the application in order to allow interoperability of the major system components such as different operating system. Multiple Operating System environment can co-exist on the same computer, in strong isolation from each other. A self-contained operating environment that behaves as if it is a separate computer

**[Smith, pg. 6, 2<sup>nd</sup> paragraph].** The virtual machine offers greater portability as well as robustness and reliability.

**As per claim 14:**

Muschellack and Smith teach the system of claim 13. Muschellack further teaches the method, wherein the resistance to tampering comprises a resistance to a change in said data **[Col. 20, lines 29-46; sealed storage]**.

**As per claim 19:**

This claim has limitations that are similar to those of claim 5, thus it is rejected with the same rationale applied against claims 5 above.

**As per claim 20:**

Muschellack and Smith teach the system of claim 13. Muschellack further teaches said policy specifies that said data is to be handled by said second software object **[fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of**

**the Trusted Platform (TP)”; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS)].**

**As per claim 23:**

Muschellack and Smith teach the system of claim 13. Muschellack further teaches said second environment is associated with a first specification that describes the behavior of said second environment, and wherein said assurance policy provides that said second environment will conform to said specification **[fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)”; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS)]; A second environment run on the RHS which relates to high-assurance is associated with specification that describe its behavior].**

**As per claim 24:**

Muschellack and Smith teach the system of claim 13. Muschellack further teaches said first environment is associated with a second specification that describes the behavior of said first environment, and wherein said first environment provides a



second level of assurance that actions performed in the first environment will be performed correctly, said second level of assurance being relatively lower than said first level of assurance [fig. 8; Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"; an application (i.e. second software object) runs on a high-assurance environment (i.e. RHS); A level of assurance of the standard mode (i.e. LHS) is relatively lower than a level of assurance of the Nexus mode (i.e. RHS)].

**As per claim 25:**

Muschellack teaches a computer-readable storage medium having stored thereon code and data to allow a user to operate on first and second types of data, said second type of data requiring a relatively higher level of protection from tampering than said first type of data, said code and data comprising:

a first software object of an application, the first software object being associated with a first specification of a first operating system, the first specification describing the behavior of said first software object, said first software object comprising instructions [fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430] to:

operate on members of said first type of data **[fig. 8; Col. 20; line 17-28; software components may continue to operate in the standard mode 430 ...];**  
recognize a member of said second type of data as not being processable by said first software object **[fig. 8; Col. 20; line 17-28, standard mode can not process data of Nexus mode];** and

cause said member of said second type of data to be routed to a second software object of the application; **[fig. 8; Col. 20, lines 24-28; “other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)”];** and

said second software object, which is associated with a second specification of a second operating system, the second specification describing the behavior of said second software object, there being a relatively higher level of assurance that said second software object will conform to said second specification than that said first software object will conform to said first specification, said second software object comprising instructions to operate on members of said second type of data **[fig. 8; Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432)],** such that functionality of the application is parsed between the first and second operating systems **[Col. 20; line 17-28; “software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which**

**have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]].**

Muschellack teaches the base layer that supports the operation of a first environment and a second environment but does not explicitly disclose a virtual machine. However, in an analogous art, Smith teaches an overview of Virtual Machine Architectures, wherein a base layer that hosts the operation of a first environment and a second environment the application [pg. 1-9; fig. 3-4].

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Smith, wherein a base layer that hosts the operation of a first environment and a second environment the application to allow interoperability of the major system components such as different operating system [Smith, pg. 6, 2<sup>nd</sup> paragraph].

**As per claim 26:**

Muschellack and Smith teach the system of claim 25. Muschellack further teaches the computer-readable medium wherein said first software object operates in a first environment, wherein said second software object operates in a second environment, wherein said first environment is associated with a third specification that describes the behavior of said first software environment, wherein said second environment is associated with a fourth specification that describes the behavior of said second environment, wherein the level of assurance that said second environment will

Art Unit: 2139

conform to said fourth specification is relatively higher than the level of assurance that said first environment will conform to said first specification, and wherein the assurance that said second software object will conform to said second specification derives from said second software object's reliance on the behavior of the second environment [fig. 8; standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432); Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ... However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]].

**As per claim 31:**

This claim has limitations that are similar to those of claim 5, thus it is rejected with the same rationale applied against claim 5 above.

**As per claim 32:**

Muschellack teaches a system that supports the partitioning of an application into at least a first software object and a second software object, the system hosting a first environment and a second environment, the first software object running in the first

environment, the second software object running in the second environment, the system comprising an application programming interface **[Col. 20; line 17-28; "software components (i.e. applications) may continue to operate in the standard mode 430 (fig. 8) or standard partition 730 (fig. 9) ...other device interface layer may continue to operate in the standard mode or partition. However, other components, such as software components which have access to secure financial information, items of value (i.e. cash, deposits) for example may operate on the nexus mode or protected partition of the Trusted Platform (TP)"]** that exposes at least one of the following methods:

(a) a first method that receives from the first software object a first data object that comprises: (1) data processable by the second software object **[Col. 20; line 17-28]**.

Muschellack does not explicitly teach a first identifier assigned by the system to the second environment; and that routes said first data object to said second environment based on said first identifier.

However, in an analogous art, Smith teaches an overview of Virtual Machine Architectures, wherein a first identifier assigned by the system to the second environment; and that routes said first data object to said second environment based on said first identifier **[pg. 1-9; fig. 3-4; Virtual machine can be used to connect the major system components; translating from one instruction set to another; optimizing an existing application binary for the same instruction set; replicating a virtual machine so that multiple (possible different) OSes can be supported**

**simultaneously; composing virtual machine software to form a more complex, flexible system"; Virtual machine controls different Operating System in different environments; The virtual machine supports multiple operating systems by assigning different identifiers and software objects in different environments].**

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack by including the teaching of Smith, wherein a first identifier assigned by the system to the second environment; and that routes said first data object to said second environment based on said first identifier in order to allow interoperability of the major system components such as different operating system. Multiple Operating System environment can co-exist on the same computer, in strong isolation from each other. A self-contained operating environment that behaves as if it is a separate computer **[Smith, pg. 6, 2<sup>nd</sup> paragraph]**. The virtual machine offers greater portability as well as robustness and reliability.

**As per claim 33:**

Muschellack and Smith teach the system of claim 32. Muschellack further teaches the system wherein said first environment is associated with a first specification that describes the behavior of said first environment, wherein said second environment is associated with a second specification that describes the behavior of said second environment, wherein there is a first level of assurance that said first environment will conform to said first specification, wherein there is a second level of assurance that said second environment will conform to said second specification, and wherein said second

level of assurance is relatively higher than said first level of assurance [fig. 8; **standard mode, Operating System; Col. 19, line 7, the standard mode or left hand side 430; Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432); It is inherent that a first specification that describes a behavior of a first environment running on the LHS (i.e. standard mode). A second specification describes a behavior of a second environment running on the RHS (i.e. Nexus mode). A level of assurance of Nexus mode is higher than a level of standard mode].**

**As per claim 34:**

Muschellack further teaches the system of claim 33, wherein said second software provides assurance that said second software object will protect data, said assurance being provided at least in part by relying on the behavior of the second environment [fig. 8, **Nexus mode, Nexus; Col. 19, line 3-7, the nexus mode (i.e. right hand side 432); applications run on a Nexus mode (i.e. high-assurance)].**

**Claims 2-4, 15-18, and 29-30** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **J.E. Smith**, "An Overview of Virtual Machine Architectures", October 27, 2001, pp. 1-20 and further in view of **Clapper** (US 2003/0107584 A1).

**As per claim 2:**

Muschellack and Smith do not explicitly teach the system of first software object causes a representation of said first of said plurality of data to be displayed on a display device, said representation comprising one or more indecipherable graphics.

However, in an analogous art, Clapper teaches a security system for visual display, wherein data is displayed on a display device, said representation comprising one or more indecipherable graphics **[fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack and Smith by including the teaching of Clapper, wherein data is displayed on a display device, said representation comprising one or more indecipherable graphics to provide secure viewing of sensitive information on a display **[Clapper; par. [0004], lines 5-6]**.

### **As per claim 3**

Clapper further teaches teaches the system of claim 2, wherein said one or more indecipherable graphics are either:

the same size as each other **[fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display]**.

### **As per claim 4:**

Muschellack and Smith do not explicitly teach a system, wherein the resistance to tampering provided by said second software object comprises said second



environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed.

However, in an analogous art, Clapper teaches a security system for visual display, wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed **[fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display; par. [0042]]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the system of Muschellack and Smith by including the teaching of Clapper, wherein the resistance to tampering provided by said second software object comprises said second environment resisting interference with the display of said first of said plurality of data by writing a representation of said first of said plurality of data into a video memory associated with a display device so as to cause said representation to supersede any image at a location on said display device at which said representation is to be displayed to provide secure viewing of sensitive information on a display **[Clapper; par. [0004], lines 5-6]**.

**As per claim 15:**

This claim has limitations that are similar to those of claim 4, thus it is rejected with the same rationale applied against claims 4 above.

**As per claim 16:**

Muschellack and Smith do not explicitly teach a first software object causes a representation of the data to be displayed on a visual display device, said representation comprising one or more indecipherable graphics.

However, in an analogous art, Clapper teaches the method of claim 13, wherein said first software object causes a representation of the data to be displayed on a visual display device, said representation comprising one or more indecipherable graphics **[fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to combine the method of Muschellack and Smith by including the teaching of Clapper, wherein data is displayed on a display device, said representation comprising one or more indecipherable graphics to provide secure viewing of sensitive information on a display **[Clapper; par. [0004], lines 5-6]**.

**As per claim 17:**

This claim has limitations that are similar to those of claim 3, thus it is rejected with the same rationale applied against claims 3 above.

**As per claim 18:**

Muschellack, Smith, and Clapper teach the method of claim 16.

Clapper further teaches wherein said first software object or said second software object, or a combination of said first software object and said second software object, cause items displayed on said visual display device to be changed in at least one respect to permit viewing of an image of the data produced by said second software object **[fig. 3-4; par. [0008]; par. [0037], lines 7-8; blurring operation to graphic data to be displayed on a display]**.

**As per claim 29:**

This claim has limitations that are similar to those of claim 3, thus it is rejected with the same rationale applied against claim 3 above.

**As per claim 30:**

This claim has limitations that are similar to those of claim 4, thus it is rejected with the same rationale applied against claims 4 above.

**Claims 21-22 and 27-28** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Muschellack et al.** (US 7,309,004 B1) in view of **J.E. Smith**, "An Overview of Virtual Machine Architectures", October 27, 2001, pp. 1-20 and further in view of **Hayman et al.** (US 5,895,966).

**As per claim 21:**

Muschellack and Smith do not explicitly teach data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed.

However, in an analogous art, Hayman teaches a security system for computer systems, wherein data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed **[abstract, fig. 3A, fig. 3B, col. 1, lines 63-64, col. 5, line 24 to col. 6, line 36; security labels are placed on each data file]**.

Therefore, it would have been obvious to the person of ordinary skill in the art at the time the invention was made to modify the method of Muschellack and Smith of the invention by including the step of Hayman, wherein data includes, or is associated with, a first label that identifies said second environment as a location in which said data is to be processed in order to provide users with a means for placing security labels on each data file or other system resource, and on each user process to enable to determine who has what type of access to which data file or other system resources **[Hayman, col. 1, line 64 to Col. 2, line 1]**.

**As per claim 22:**

Muschellack, Smith, and Hayman teach the method of claim 21. Hayman further teaches said data includes, or is associated with, a second label that identifies said

second software object as a processor for said data, and wherein said second environment routes said data to said second software object based on said second label **[abstract, fig. 3A, fig. 3B, col. 1, lines 63-64, col. 5, line 24 to col. 6, line 36; security labels are placed on each data file]**.

**As per claim 27:**

This claim consists the computer-readable medium wherein each member of said second type of data to implement claim 21, thus it is rejected with the same rationale applied against claim 21 above.

**As per claim 28:**

This claim consists the computer-readable medium wherein said first software object causes said member of the second type to be routed to said second software object by sending said member of the second type to a base component, said first label being assigned by said base component, said second label being recognizable by said second environment and not by said base component to implement claim 22, thus it is rejected with the same rationale applied against claim 22 above.

***Conclusion***

The prior arts made of record and not relied upon are considered pertinent to applicant's disclosure.

US 6192477 B1 Corthell; David;

US 6327652 B1 to England; Paul et al.;

US 20020124177 A1 to Harper, Travis Kelly et al.;

US 7082507 B1 to Christie; David S. et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Canh Le whose telephone number is 571-270-1380.

The examiner can normally be reached on Monday to Friday 7:30AM to 5:00PM other Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kristine Kincaid can be reached on 571-272-4063. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Canh Le/

Examiner, Art Unit 2139

April 23, 2008

Application/Control Number: 10/693,749  
Art Unit: 2139

Page 30

/Kristine Kincaid/  
Supervisory Patent Examiner, Art Unit 2139